In [ ]:
```
import pandas as pd
```

In [ ]:
```
r_cols = ['user_id', 'movie_id','rating', 'timestamp']
```

In [ ]:
```
ratings = pd.read_csv("u.data", sep = "\t", names = r_cols)
```

In [ ]:
```
ratings.head()
```

Out[ ]:

	user_id	movie_id	rating	timestamp
0	196	242	3	881250949
1	186	302	3	891717742
2	22	377	1	878887116
3	244	51	2	880606923
4	166	346	1	886397596

In [ ]:
```
ratings[ratings['user_id']==196]
```

Out[ ]:

	user_id	movie_id	rating	timestamp
0	196	242	3	881250949
940	196	393	4	881251863
1133	196	381	4	881251728
1812	196	251	3	881251274
1896	196	655	5	881251793
2374	196	67	5	881252017
6910	196	306	4	881251021
7517	196	238	4	881251820
7842	196	663	5	881251911
10017	196	111	4	881251793
10254	196	580	2	881252056
10981	196	25	4	881251955
13733	196	286	5	881250949
14606	196	94	3	881252172
16834	196	692	5	881252017
17102	196	8	5	881251753
17830	196	428	4	881251702
18853	196	1118	4	881252128
21605	196	70	3	881251842
22271	196	66	3	881251911
22773	196	257	2	881251577
23189	196	108	4	881252110
24030	196	202	3	881251728
25726	196	340	3	881251045
32721	196	287	3	881251884
33536	196	116	3	881251753
35197	196	382	4	881251843
36281	196	285	5	881251753
41539	196	1241	3	881251642
42384	196	1007	4	881251601
50147	196	411	4	881252090
52726	196	153	5	881251820
56628	196	13	2	881251955
59165	196	762	3	881251955
59607	196	173	2	881251820
60199	196	1022	4	881251143
60706	196	845	4	881251954
78787	196	269	3	881250949
87863	196	110	1	881252305

In [ ]:
```
### 196 user rated 39 movies
```

In [ ]:
```
ratings[ratings['movie_id']==242]
```

Out[ ]:

	user_id	movie_id	rating	timestamp
0	196	242	3	881250949
253	63	242	3	875747190
629	226	242	5	883888671
1232	154	242	3	879138235
2159	306	242	5	876503793
...	...	...	...	...
95720	305	242	5	886307828
96237	845	242	4	885409493
97046	500	242	3	891916883
99260	720	242	4	891262608
99759	721	242	3	877137597

117 rows × 4 columns

In [ ]:
```
## 117 people given the rating for movie 242
```

In [ ]:
```
ratings['user_id'].nunique()
```

Out[ ]: 943

In [ ]:
```
ratings['movie_id'].nunique()
```

Out[ ]: 1682

In [ ]:
```
ratings.shape
```

Out[ ]: (100000, 4)

In [ ]:
```
# 943 people rated 1682 movies and the total number of ratings given are 100000
```

In [ ]:
```
### knn basic and another one SVD (singular value decomposition)
```

In [ ]:
```
### surprise library
```

In [ ]:
```
!pip install surprise
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting surprise
  Downloading surprise-0.1-py2.py3-none-any.whl (1.8 kB)
Collecting scikit-surprise
  Downloading scikit-surprise-1.1.3.tar.gz (771 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 772.0/772.0 KB 13.3 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: joblib>=1.0.0 in /usr/local/lib/python3.9/dist-packages (from scikit-surprise->su
rprise) (1.1.1)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.9/dist-packages (from scikit-surprise->su
rprise) (1.22.4)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.9/dist-packages (from scikit-surprise->sur
prise) (1.10.1)
Building wheels for collected packages: scikit-surprise
  Building wheel for scikit-surprise (setup.py) ... done
  Created wheel for scikit-surprise: filename=scikit_surprise-1.1.3-cp39-cp39-linux_x86_64.whl size=3193673 sha2
56=454dfbf0e7591d91ee0f6028b378d8e3dbed231e231eed0ceba53c32863c865c
  Stored in directory: /root/.cache/pip/wheels/c6/3a/46/9b17b3512bdf283c6cb84f59929cdd5199d4e754d596d22784
Successfully built scikit-surprise
Installing collected packages: scikit-surprise, surprise
Successfully installed scikit-surprise-1.1.3 surprise-0.1
```

In [ ]:
```
import surprise
```

In [ ]:
```
## user_id, movie_id and rating
```

In [ ]:
```
ratings = ratings.drop("timestamp",axis=1)
```

In [ ]:
```
ratings
```

Out[ ]:

	user_id	movie_id	rating
0	196	242	3
1	186	302	3
2	22	377	1
3	244	51	2
4	166	346	1
...	...	...	...
99995	880	476	3
99996	716	204	5
99997	276	1090	1
99998	13	225	2
99999	12	203	3

100000 rows × 3 columns

In [ ]:
```
### reader object
## reader object hepls in parsing the dataframe for ratings
```

In [ ]:
```
from surprise import Reader, Dataset
```

In [ ]:
```
reader = Reader()
```

In [ ]:
```
data = Dataset.load_from_df(ratings, reader)
```

In [ ]:
```
data
```

Out[ ]:
```
<surprise.dataset.DatasetAutoFolds at 0x7f49af8c9820>
```

In [ ]:

SVD

In [ ]:
```
from surprise import SVD
```

In [ ]:
```
algo = SVD()
```

In [ ]:
```
from surprise.model_selection import cross_validate
```

In [ ]:
```
cross_validate(algo, data, measures= ["RMSE"], cv = 5 , verbose = True)
```

```
Evaluating RMSE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.9392  0.9434  0.9350  0.9342  0.9308  0.9365  0.0043
Fit time          1.11    1.11    1.10    1.09    1.09    1.10    0.01
Test time         0.28    0.16    0.22    0.12    0.24    0.20    0.06
```

Out[ ]:
```
{'test_rmse': array([0.9392463 , 0.94336732, 0.93503498, 0.93419255, 0.93081398]),
 'fit_time': (1.1077826023101807,
  1.1129565238952637,
  1.1049079895019531,
  1.0867764949798584,
  1.0940277576446533),
 'test_time': (0.27699780464172363,
  0.156691312789917,
  0.22157621383666992,
  0.12337613105773926,
  0.23676347732543945)}
```

In [ ]:
```
from surprise import KNNBasic
```

In [ ]:
```
knn_algo = KNNBasic()
```

In [ ]:
```
cross_validate(knn_algo, data, measures= ["RMSE"], cv = 5 , verbose = True)
```

```
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Evaluating RMSE of algorithm KNNBasic on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.9779  0.9748  0.9876  0.9823  0.9714  0.9788  0.0057
Fit time          0.28    0.24    0.26    0.27    0.22    0.26    0.02
Test time         3.58    3.12    3.14    3.70    3.16    3.34    0.25
```

Out[ ]:
```
{'test_rmse': array([0.9778934 , 0.97477033, 0.98763122, 0.98226676, 0.97137762]),
 'fit_time': (0.28258538246154785,
  0.2387838363647461,
  0.25832080841064453,
  0.2730586528778076,
  0.2225339412689209),
 'test_time': (3.584620714187622,
  3.115795373916626,
  3.137979507446289,
  3.697896718978882,
  3.156301259994507)}
```

In [ ]: